Laboratorio 5: "Tarjeta de Desarrollo Spartan-3"

Objetivos:

- Usar la interfaz serial sincrónica PS/2 como dispositivo de entrada de datos.
- Diseñar un multiplexor con división de tiempo para manejo de display de 7 segmentos.
- Desarrollar métodos de prueba para módulos individuales y el circuito completo.

Tarjeta de Desarrollo Spartan-3 de Digilent

En este laboratorio se utilizarán algunos recursos internos que dispone la tarjeta de desarrollo con el fin de aprender su funcionamiento. Específicamente se utilizarán como entrada de datos la interfaz PS/2, los pulsadores e interruptores, y como salida se usarán leds y displays de 7 segmentos. Para cada uno de estos recursos se diseñarán pequeños módulos que permitan aprender su funcionamiento.

La interfaz PS/2 que dispone la tarjeta de desarrollo corresponde al estándar IBM y será presentada con más detalle en la próxima sección. En el sitio del curso se ha dispuesto de material de apoyo adicional.

Además, la tarjeta dispone de 8 leds y 4 displays de 7 segmentos como salida de datos. Los leds se activan con un 1 lógico. Los segmentos de los displays se activan con un 0 lógico. Para reducir la cantidad de pines necesarios para la activación de los displays de 7 segmentos, éstos se encuentran conectados en paralelo. La activación de un display se logra mediante un pin de control conectado al ánodo común del display (mediante un 0 lógico conectado a la base de un transistor que alimenta al display). Con esto se logra reducir la cantidad de salidas requeridas de 36 a 12. Para poder observar varios dígitos encendidos (pseudo-) simultáneamente, es necesario multiplexar en el tiempo las señales de control, refrescando continuamente los displays. Una descripción detallada se encuentra en el capítulo 3 del manual del usuario.

Interfaz PS/2

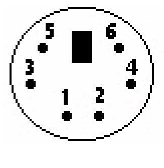
La interfaz de comunicación PS/2 es una interfaz propuesta por IBM para comunicar dispositivos seriales en forma sincrónica, tanto teclado como Mouse. Las principales características de un teclado PS/2 son:

- Gran número de teclas (101 a 104).
- Conector de 5 o 6 pines (incluyen adaptadores).
- Protocolo de comunicación serial bidireccional (PS/2).
- Garantizan sólo el conjunto 2 de los *scan codes*.
- Responden a todos los comandos enviados, sin embargo no actúan en todos ellos.

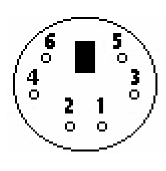
La interfaz física que usualmente se utiliza se muestra en la Figura 1.

La alimentación del teclado/mouse corresponde a Vcc = +5 [V] y una corriente máxima de 100 [mA]. Se recomienda no conectar el teclado a la tarjeta mientras se encuentre encendida. Las líneas de Data y Clock son ambas de colector abierto, con resistencias de "pull-up" para fijar ambas líneas en alto.

El mouse y teclado PS/2 implementan un protocolo de comunicación serial bidireccional. El bus de comunicación se encuentra en estado "*idle*" cuando ambas líneas (data y clock) se encuentran en alto. Este es el único estado en que al dispositivo (teclado o mouse) le está permitido enviar información al *host* (computador o en nuestro caso tarjeta de desarrollo Spartran 3). El *host* tiene el control último sobre el bus y puede inhibir la comunicación en cualquier instante, colocando la línea de clock en nivel bajo.



Plug



Socket

6-pin Mini-DIN (PS/2):
1 - Data
2 - No Implementado
3 - Ground
4 - Vcc (+5V)
5 - Clock
6 - No Implementado

Figura 1: Interfaz PS/2

El dispositivo siempre genera la señal de clock. Si el *host* desea enviar datos, debe primero inhibir la comunicación desde el dispositivo, colocando la línea de clock en nivel bajo. Luego debe colocar en nivel bajo la línea de data y subir la línea de clock. Este estado es conocido como "*Request to Send*", con lo cual el *host* señaliza al dispositivo que comience a generar pulsos de reloj a través de la línea clock para enviar los datos. Por lo tanto el bus puede estar en uno de los siguientes tres estados:

- **IDLE**: Data y Clock en alto
- **INHIBIT**: Data en alto y Clock en bajo
- **REQUEST TO SEND**: Data en bajo y Clock en alto

Todos los datos son enviados de un byte por vez y cada byte es enviado dentro de un frame de 11 o 12 bit:

- 1 bit de partida: siempre es 0.
- 8 bit de datos: se comienza por el menos significativo.
- 1 bit de paridad: se utiliza paridad impar.
- 1 bit de parada: siempre es 1.
- 1 bit de confirmación: sólo para comunicación desde el *host* al dispositivo.

El bit de paridad es colocado en alto si hay un número par de 1's en los bit de datos, y colocado en bajo si hay un número impar. La idea es que el número de 1's de los bit de datos más el bit de paridad siempre sean un número impar (paridad impar). Esto se utiliza para la detección de errores en la transmisión. Si el dispositivo detecta un error responde como que se le ha enviado un comando inválido. La Figura 2 muestra la comunicación del dispositivo con el *host*.

Cuando el teclado o mouse desean mandar información deben asegurarse que la línea de clock debe estar en alto por lo menos 50 [us] antes de que el dispositivo pueda comenzar a enviar datos. El teclado/mouse escribe un bit en la línea de data cuando el reloj está alto, y es leído en el *host* cuando el reloj está bajo. Como se trata de una comunicación desde el dispositivo hacia el *host* no se envía un bit de confirmación.

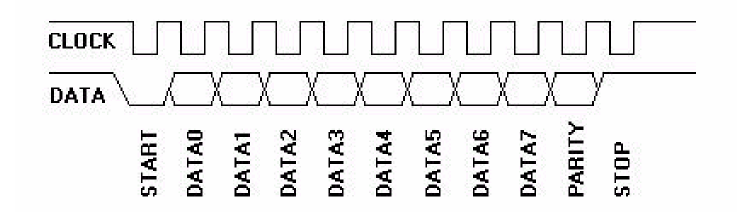


Figura 2: Comunicación PS/2 - *Host*

Scan Codes

En los teclados se distinguen distintos conjuntos de *scan codes* para identificar la tecla o teclas que se presionan. Se utiliza por omisión el conjunto 2. Cada tecla tiene asociado un *scan code* compuesto de dos códigos: un *make code*, que se emite cada vez que se presiona una tecla, y un *break code*, que se emite cuando se suelta la tecla. En general estos códigos suelen tener entre 1 y 2 byte, pero existen ciertos *scan codes* que son más largos debido a que corresponden a una combinación de teclas más elaborada.

En http://www.computer-engineering.org/ps2keyboard/scancodes2.html se encuentra disponible los *scan codes* del conjunto 2, así como también en el manual de referencia de la tarjeta de desarrollo capítulo 6, disponible en el sitio de la asignatura.

Driver PS/2

En el sitio de la asignatura se encuentra disponible un driver PS/2 escrito en lenguaje Verilog (`kbd.v`). La Figura 3 muestra la idea principal del funcionamiento del driver.

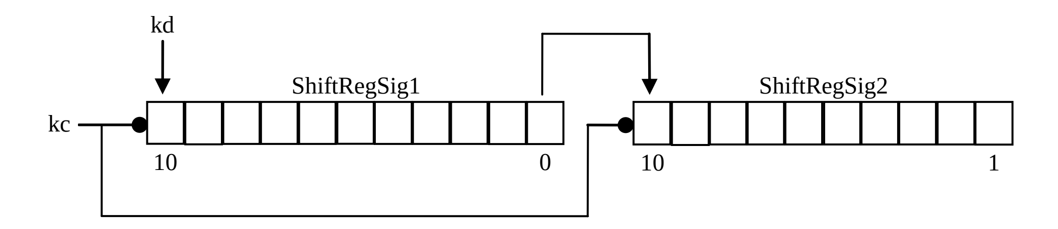


Figura 3: Conexión Lógica para la Captura de Datos en Serie

El driver de teclado realiza la captura de datos en serie vía la línea `kd`, para lo que utiliza 2 registros de desplazamiento de 11 y 10 bit respectivamente. El reloj de sincronismo `kc` proviene de la misma fuente de los datos y está conectado a ambos registros. El bit más significativo del registro 1 es el que recibe los datos seriales. Al capturar el valor hexadecimal "`F0`" en el registro 2, se ha recibido el *break code* de una tecla, cuyo valor está almacenado en el registro 1 entre los bit 8 al 1.

P5. Actividades Previas

P5.1 Contador BCD

En el sitio de la asignatura se encuentra el código en Verilog de un conversor binario a 7 segmentos (ssdec.v). Este código tiene como entradas el dato en binario de 4 bit, el estado de punto decimal y el tipo de dispositivo (ánodo o cátodo común). La salida del módulo es el estado de cada segmento del display, incluido el punto decimal. Debido a la configuración de los display 7 segmentos en la tarjeta de desarrollo, para poder observar varios dígitos encendidos será necesario multiplexar en el tiempo las señales de control, refrescando continuamente el display. Complete el módulo `display()` **respetando** las siguientes características de entrada y salida:

```
module display(clk, num, ssg, dctl);
input clk;
input [15:0] num;
output [7:0] ssg;
output reg [3:0] dctl;

   // fill in

endmodule
```

donde `clk` es una señal de reloj con una frecuencia tal que permita multiplexar el control del display en el tiempo, `num` es el valor de entrada de 16 bit, `ssg` y `dctl` corresponden a las salidas de los segmentos, incluido el punto, y las señales de control para el encendido del display.

Un contador BCD facilita la visualización decimal de cuentas binarias. El siguiente módulo es un contador en BCD de 4 dígitos:

```
module bcdcounter(rst, clk, value);
input rst, clk;              // reset, clock
output reg [15:0] value;     // 4 bcd digits

   always @(negedge clk or posedge rst) begin
      if (rst == 1)
         value <= 0;
      else begin
         if (value[3:0] == 9) begin
            value[3:0] <= 0;
            if (value[7:4] == 9) begin
               value[7:4] <= 0;
               if (value[11:8] == 9) begin
                  value[11:8] <= 0;
                  if (value[15:12] == 9) value[15:12] <= 0;
                  else value[15:12] <= value[15:12] + 1;
               end else value[11:8] <= value[11:8] + 1;
            end else value[7:4] <= value[7:4] + 1;
         end else value <= value + 1;
      end
   end

endmodule
```

Utilizando los módulos anteriores, diseñe un módulo que permita la visualización en el display del tiempo transcurrido desde que se presionó un pulsador (cualquiera de los 4 pulsadores) en formato MM.SS. Construya un módulo de prueba para simular el diseño, incluyendo las secciones de inicialización que sean necesarias (pero que no son relevantes para la síntesis).

P5.2 Uso de Pulsadores, Interruptores y Leds

Diseñe un módulo que permita asociar a cada interruptor de la tarjeta de desarrollo un led para desplegar el estado del interruptor respectivo. Añada la posibilidad de encender, apagar, invertir y hacer una rotación hacia la derecha del estado de los leds al presionar el pulsador 0, 1, 2 y 3 respectivamente. La cantidad a rotar dependerá de la cantidad de interruptores activos del momento. Ej, sea `swt` = 0x15 el estado de los interruptores, es decir interruptores 0, 2, y 4 están activos, el pulsador 0 muestra led = 0xff, pulsador 1 muestra led = 0x0, el pulsador 2 muestra led = 0xEA y el pulsador 3 muestra 0xA2.

Construya un módulo de prueba para verificar el diseño mediante simulación.

P5.3 Interfaz PS/2

Utilizando el módulo `display()` desarrollado en el punto P5.1 y el driver PS/2, diseñe un módulo que muestre el *scan code* de la tecla presionada en los dígitos menos significativos del display y la interpretación de la tecla en los dígitos más significativos para al menos 10 teclas (e.g. tecla 1 despliega: 0116. Si es necesario, construya módulos de prueba para verificar su diseño.

L5. En el Laboratorio.

L5.1. Revisión Actividades Previas

Muestre al profesor o ayudantes la simulación funcional y temporal de los módulos desarrollados en los puntos P5.1 y P5.2. Muestre al profesor el diseño del módulo del punto P5.3 que usa el driver PS/2.

L5.2. Contador BCD

Sintetice el diseño del reloj BCD del punto P5.1 y verifique su funcionamiento.

L5.3. Uso de Pulsadores, Interruptores y Leds

Sintetice el diseño del punto P5.2 y verifique su funcionamiento.

L5.4. Interfaz PS/2

Sintetice el diseño del punto P5.3 y verifique su funcionamiento.

L5.5. Calculadora Básica

Diseñe una calculadora básica usando notación postfija (eg. 4↵2↵+ con '↵' la tecla enter), con las operaciones de suma, resta y multiplicación para números hexadecimales sin signo de 16 bit. La entrada de datos es vía teclado. Estos números ingresan en la posición menos significativa del número, posterior a un corrimiento del operando en 4 bit hacia la izquierda (e.g. al presionar la tecla '4' seguida de la tecla '5', se visualiza el número 0045 en el display). Usando pulsadores, implemente las funciones CE (Clear Everything), C (Clear), para borrar todos los registros y borrar el último operando respectivamente. No se preocupe de rebalses ni de los signos de los operandos. El display debe desplegar en todo momento el último operando ingresado ó el resultado de la última operación. Al presionar otro pulsador, el display muestra el segundo operando o 0 si no se ha ingresado aún.

L5.6. Contadores, Interruptores y Leds

Al circuito de la Calculadora Básica, añada tres contadores en BCD para indicar la cantidad de veces que se ha ejecutado la operación de suma, resta y multiplicación. La cuenta se muestra en el display según la posición de dos interruptores. Debe incluir para cada contador un *reset* de la cuenta al presionar un pulsador (distinto al usado para CE o C). Finalmente, utilice los leds para: mostrar las señales CE y C, para indicar que lo que se muestra en el display es el resultado de una operación de suma, resta o multiplicación, el estado de los interruptores, y la solicitud de *reset* de una cuenta.